Math 151b: Problem Set 6

Owen Jones

2/23/2024

Problem 1
Suppose a solution $u(x)$ exists.
We use the divergence theorem: $\int_a^b u''(x)dx = u'(b) - u'(a)$.
We are given $u''(x) = f(x)$, $u'(a) = \alpha$, and $u'(b) = \beta$. Substituting into our equation, we obtain $\int_a^b f(x)dx = \beta - \alpha$.
The solution to the Neumann problem is not unique because it does not take into account the absolute temperature of the bar. $u(x)$ and $u(x) + c$ can both be solutions to the problem, but one solution can have a higher absolute temperature at every point on the interval.
Problem 2
$u(x_{j+1}) = u(x_j + h) = u(x_j) + hu'(x_j) + \frac{h^2}{2}u''(x_j) + \frac{h^3}{6}u'''(x_j) + O(h^4)$
$u(x_{j-1}) = u(x_j - h) = u(x_j) - hu'(x_j) + \frac{h^2}{2}u''(x_j) - \frac{h^3}{6}u'''(x_j) + O(h^4)$
The zeroth, first, and third derivatives cancel out leaving
$\frac{u(x_{j+1})-2u(x_j)+u(x_{j-1})}{h^2} = \frac{h^2u''(x_j)+O(h^4)}{h^2} = u''(x_j) + O(h^2)$.
Problem 3

(a) Because $\mathbf{x}$ is a vector s.t $A\mathbf{x} = \mathbf{0}$ $\sum_{j=1}^{n} a_{ij}x_j = 0$ $\forall i = 1 \ldots n$. Fix some i.

It follows $a_{ii}x_i = -\sum_{j=1,j\neq i}^{n} a_{ij}x_j$. Taking the absolute value of both sides,

we obtain $|a_{ii}||x_i| = |a_{ii}x_i| = |-\sum_{j=1,j\neq i}^{n} a_{ij}x_j| = |\sum_{j=1,j\neq i}^{n} a_{ij}x_j|$.

Thus, $|a_{ii}||x_i| = |\sum_{j=1,j\neq i}^{n} a_{ij}x_j|$.

(b) Applying the Triangle Innequality to the RHS of (2), we obtain
$|\sum_{j=1,j\neq i}^{n} a_{ij}x_j| \leq \sum_{j=1,j\neq i}^{n} |a_{ij}||x_j|$.
Using (3) i.e $|x_i| = \max_{1\leq j\leq n} |x_j|$, we obtain

$$\sum_{j=1, j\neq i}^{n} |a_{ij}||x_j| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}||x_i|.$$

Together with (1) $\displaystyle\sum_{j=1, j\neq i}^{n} |a_{ij}| < |a_{ii}|$ we obtain

$$|\sum_{j=1, j\neq i}^{n} a_{ij}x_j| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}||x_i| < |a_{ii}||x_i|$$

which is a contradiction because we claimed $\displaystyle|\sum_{j=1, j\neq i}^{n} a_{ij}x_j| = |a_{ii}||x_i|$.

Thus, A must be non-singular.

Problem 4

(a) The central difference formula for $u''(x_i) \approx \frac{u(x_{i+1}-2u(x_i)+u(x_{i-1}))}{h^2}$.
Thus, $f(x_i) = \frac{u(x_{i+1}-2u(x_i)+u(x_{i-1}))}{h^2} - cu(x_i) = \frac{u(x_{i+1}-(2+ch^2)u(x_i)+u(x_{i-1}))}{h^2}$.
Let $\mathbf{u} = [u_1, u_2, \ldots, u_{N-1}]^\top$ and $\mathbf{f} = [f_1, f_2, \ldots, f_{N-1}]^\top$ where $u_i = u(x_i)$ and $f_i = f(x_i)$.

(b) At the boundaries we have $\frac{\alpha-(2+ch^2)u_1+u_2}{h^2} = f_1 \Rightarrow \frac{-(2+ch^2)u_1+u_2}{h^2} = f_1 - \frac{\alpha}{h^2}$ and $\frac{u_{N-2}-(2+ch^2)u_{N-1}+\beta}{h^2} = f_{N-1} \Rightarrow \frac{u_{N-2}-(2+ch^2)u_{N-1}}{h^2} = f_{N-1} - \frac{\beta}{h^2}$.
Thus, our matrix $A : (N-1) \times (N-1)$ can be represented as

$$A = \frac{1}{h^2}\begin{bmatrix} -(2+ch^2) & 1 & 0 & 0 & \cdots & 0 \\ 1 & -(2+ch^2) & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 1 & -(2+ch^2) \end{bmatrix}$$

giving us $A\mathbf{u} = \mathbf{f} - [\frac{\alpha}{h^2}, 0, \ldots, 0, \frac{\beta}{h^2}]^\top$
Fix some row i.

$$\sum_{j=1, j\neq 0}^{N-1} |a_{ij}| = |\frac{1}{h^2}| + |\frac{1}{h^2}| = \frac{2}{h^2} < \frac{2+ch^2}{h^2} = |-\frac{2+ch^2}{h^2}| = |a_{ii}|,$$

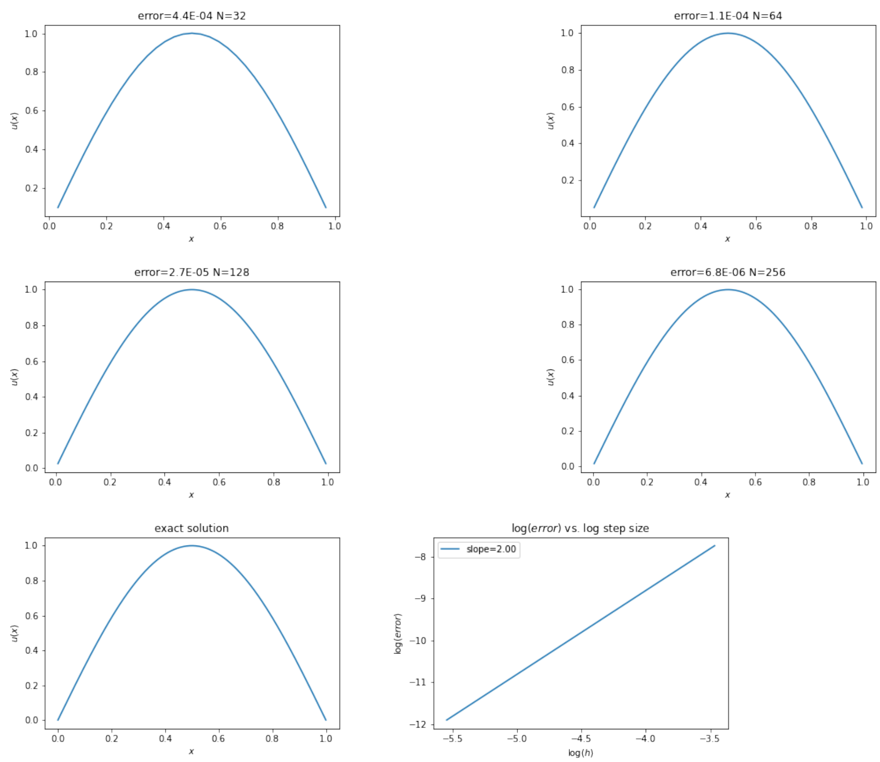
so $\displaystyle\sum_{j=1, j\neq 0}^{N-1} |a_{ij}| < |a_{ii}| \Rightarrow A$ is strictly diagonally dominant, hence invertible.

Problem 5

(a) Choose $u(x) = \sin(\pi x)$ with $c = 3$. Thus, $f(x) = -(\pi^2+3)\sin(\pi x)$.

(b) $u(0) = u(1) = 0 \Rightarrow \alpha = \beta = 0$

(c) Below shows comparison between numerical and exact solution for $N = 32, 64, 128$, and 256 points with error. Numerical solutions closely approximate the exact solution, so it is pretty difficult to distiguish between them. Plotted the log of the errors for each of the numerical solutions against the log of the step size to determine order of accuracy of the approximation. Because the slope of the line is 2, it follows $error = O(h^2)$



In [1]:

```
import numpy as np
from matplotlib import pyplot as plt
from math import pi
```

In [2]:

```
a=0
b=1
c=3
numpts=32
```

In [3]:

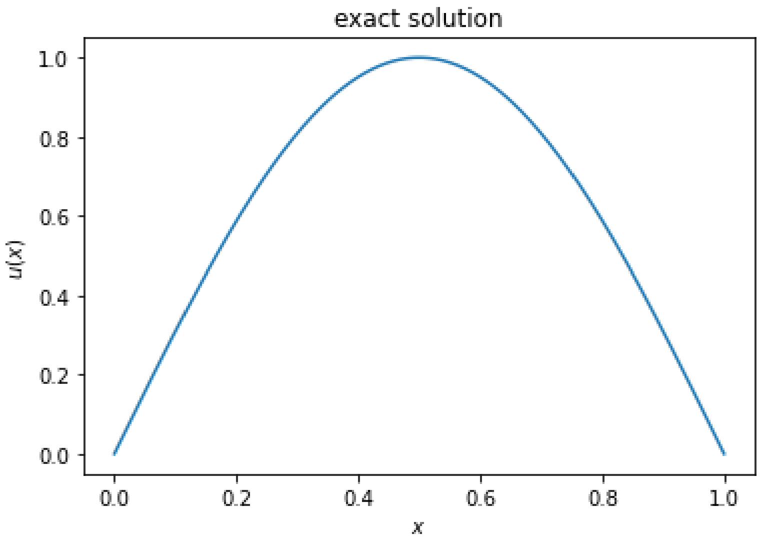
```
f = lambda x: -1*(c+pi**2)*np.sin(pi*x)
u_ex = lambda x: np.sin(pi*x)
```

In [4]:

```
def BVP(a,b,u,f,numpts,plot=True):
    alpha=u(a)
    beta=u(b)
    xvec=np.linspace(a,b,numpts+1)
    h=xvec[1]-xvec[0]
    Amat=(np.identity(numpts-1)*(-2-c*h**2)+np.diag(np.ones(numpts-2),k=1)+np.di
    v=np.array([np.append(np.insert(np.zeros(numpts-3),0,alpha),beta)]).transpos
    bvec=np.array([f(xvec[1:-1])]).transpose()-v
    uvec=np.matmul(np.linalg.inv(Amat),bvec)
    u_ext=np.array([u(xvec[1:-1])]).transpose()
    err=(h**0.5)*np.linalg.norm(uvec-u_ext,ord=2)
    if plot==True:
        plt.plot(xvec[1:-1],uvec)
        #plt.plot(xvec[1:-1],u_ext)
        plt.title(f'error={err:.1E} N={numpts}')
        plt.xlabel('$x$')
        plt.ylabel('$u(x)$')
        plt.savefig(f'hw_6_q_5_N_{numpts}')
    return err
```

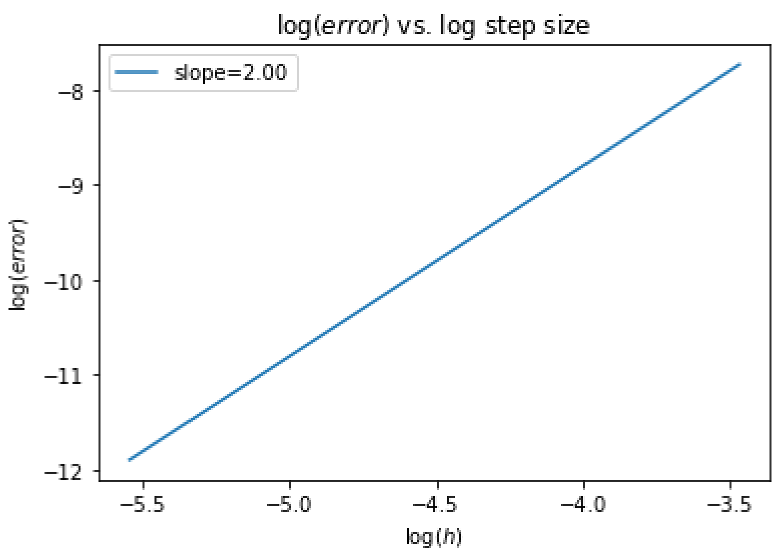
In [5]:

```
error=np.array([])
numpt_array=np.array([32,64,128,256])
for i in numpt_array:
    error =np.append(error,BVP(0,1,u_ex,f,i))
    plt.clf()
plt.plot(np.linspace(0,1,257),u_ex(np.linspace(0,1,257)))
plt.title('exact solution')
plt.xlabel('$x$')
plt.ylabel('$u(x)$')
plt.savefig('exact_solution_hw_6_q_5')
```



In [25]:

```
fit=np.polyfit(np.log(1/numpt_array),np.log(error),deg=1)
plt.plot(np.log(1/numpt_array),np.log(error),label=f"slope={fit[0]:.2f}")
plt.xlabel('$\log(h)$')
plt.ylabel('$\log(error)$')
plt.title('$\log(error)$ vs. log step size')
plt.legend()
plt.savefig('q_5_error_vs_step_size')
```



In [ ]:

```
```